# Datasets and Event Collections

ATLAS software meeting

Database session

David Adams

BNL

November 19, 2002

# Contents

- Introduction
- Event collection goals
- Dataset goals
- Analysis use case
- Dataset requirements
- Dataset types
- Dataset interface
- Event collection interface
- Integration

# Introduction

There is a need to to deal with collections of event data at more that just the file level

LCG POOL includes event collections

- ODB-like interface
  - Event collection is a list of refs to event header objects
  - Event header is a list of refs to data objects
- Less natural for HES (hybrid event store)
  - Workable if refs include file ID
  - But this leads to problems with object replication
    - > Object changes identity when copied to a new file
    - > Or connection between ref and new file ID requires catalog

# Introduction (cont)

Here we introduce datasets

- Also provide a collective view of event data
- Provide explicit connection to files
  - More natural for HES
- Use distributed analysis use case to generate requirements and interface
- Compare and integrate with event collections
  - Make use of existing design and code
  - Maintain connection to tag metadata for rapid event selection

# Event collection goals

Identify a collection of event data

- For physics analysis, calibration, alignment,…

Provide means to iterate over the events

Provide means to associate each event with metadata (attributes)

- Metadata in RDB to allow rapid queries

# Dataset goals

Identify a collection of event data

- For physics analysis, calibration, alignment…
- Might also want non-event data
  - e.g. relevant conditions data

Provide information for locating that data

- E.g. set (or sets) of logical file names
- May account for event or content selection

Serve as input for distributed analysis

Serve a input and output for distributed production

- E.g. data units in a virtual data system

# Analysis use case

User locates an existing dataset

- From physics group web page or
- Query on dataset catalog

User applies event filter to this dataset

- Based on tag metadata
- Could be an event ID list

User specifies a task to apply to each event

- E.g. fill a histogram or event ID list
- Specification includes the content (e.g. “refit tracks”) required as input to the task

# Analysis use case (cont)

User specifies the application to apply this task

- E.g. athena

User submits dataset, task and application to a scheduler

Scheduler applies content filter to the dataset

- Content specified by the task
- User might do this before submission

Scheduler locates the data (files) included in dataset

- Might be at another site and
- Distributed over multiple sites

# Analysis use case (cont)

Scheduler divides dataset into sub-datasets

- Size appropriate for job submission
- Division along file and location boundaries

Scheduler creates a job for each sub-dataset

- Application and task
- Site, farm or node

Scheduler returns estimate of job cost to user

- Clock time, CPU, data transfer

User confirms request

Scheduler submits and monitors jobs

# Analysis use case (cont)

Each job is run independently

- Files defined by the dataset are staged
  - Or better, job was assigned where files are already staged
- Application is used to define executable
- Task is compiled and dynamically linked
- Executable loops over events in the dataset
  - Event data objects in dataset are loaded
  - Task is run for each event updating results
- Concatenated result is made available to the scheduler

# Analysis use case (cont)

As jobs complete, scheduler concatenates results

Partial result is available to user

- Includes concatenation of the processed sub-datasets

Scheduler notifies the user when the complete result is available

User fetches the result

- Histograms are displayed
- Event ID list may be used as event filter to create a new dataset for use as input in the next analysis step

# Dataset requirements

The preceding use case is used to deduce the following requirements for datasets:

Events

- Dataset has a data for a well-defined list of event ID's
- There are means to select events
  - I.e. use a dataset to create a new dataset with a subset of the events in the original

# Dataset requirements (cont)

## Content

- Each event has a well-defined content (tracks, jets,…)
- Consistent dataset has the same content for all events
  - We can speak of the content of the dataset
  - Require consistency except some content may be missing in some events
- Natural in ATLAS to use something like StoreGate type-keys to label content

# Dataset requirements (cont)

## Partitioning

- It is possible to partition a dataset into sub-datasets
- Along event or content boundaries
- Sub-dataset is a dataset

## Concatenation

- It is possible to concatenate datasets
- Different events with the same content
  - (more precisely, different event ID's)
- Or different content for the same events
- Concatenated dataset is a dataset

# Dataset requirements (cont)

## Files

- Where data resides in a file (usual case), there are means to discover the complete set of files holding this data
  - Usually by logical file name
- May be multiple sets of files (object replication)

## Data access

- Means are provided to the fill transient store
  - One event at a time
  - Content defined by the dataset
  - Data taken from the files

# Dataset requirements (cont)

## Provenance

- Dataset(s) from which the current dataset was derived
- Transformation used in this derivation
  - Sequence of algorithms
  - Selection criteria
- Sufficient to reproduce dataset
- Each dataset and transformation has a unique ID to enable recording
- Might be expressed in the context of a virtual data catalog

# Dataset requirements (cont)

## Catalogs

- Datasets are cataloged with metadata so users can discover datasets of interest
- Metadata includes
  - event ID's
  - content
  - production and selection history (provenance)
  - data location (file sets)
- Not necessary to catalog all partitions and concatenations
  - these can be in inferred

# Dataset types

There are different types of datasets:

- Collection of object identifiers indexed by event ID and type-key
  - ODB-like event store required to access data
  - Similar to explicit event collection
- Self-describing file or list of such files
  - Organization of data in files allows indexing by event ID and type-key
  - Similar to implicit event collection
- Concatenation of datasets
- Event and/or content selection on another dataset

# Dataset interface

Users access any of these types through a common interface that provides the following:

- Event ID list
- Content list (e.g. type-keys)
- Means to access data object associated with an event ID and content ID
- Set or sets of files holding the data
- Partitioning along file boundaries
- Provenance
- Event iterator providing optimal access for serial files

# Event collection interface

Event collections provide the following interface:

- Iterator over event headers
- Means to access the data objects listed in each event header
  - Access through POOL event store
- Event selection based on tag data (attributes)
  - Only for explicit collections

# Integration

Dataset model adds new and important functionality.

Here is a proposal for integration datasets and event collections:

- Use dataset interface as basis for input to processing (athena) and job schedulers
- Use existing explicit event collection to implement the corresponding dataset type
- Add event ID to tag data so that selections can be made for all types of datasets